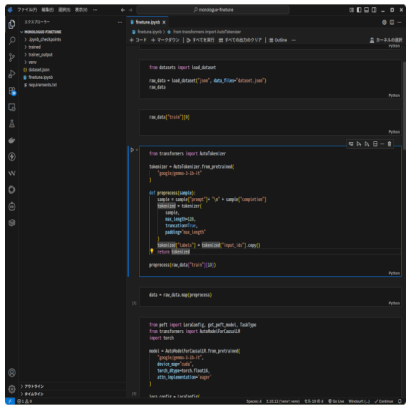


?????????????? LLM????????????

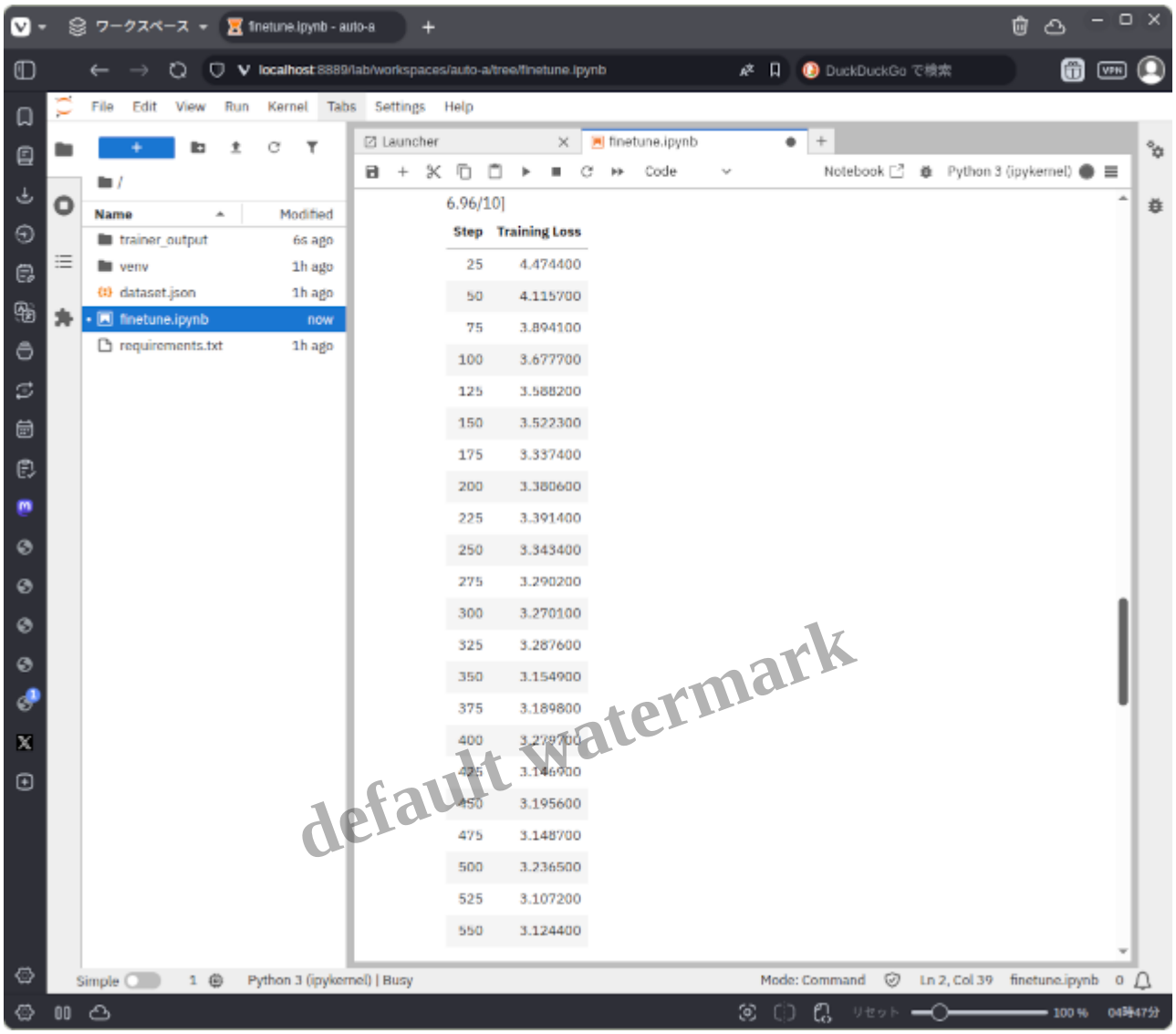
Description

??Pre
Language Model??Lora??????????????????????????????????????

????????????Core i5 13500 CPU?NVIDIA RTX A4000 (16GB VRAN)??????????????????????Google?Gemma 3 1B??????????????????????????

????????????????????????????????

default watermark



default watermark

??VRAM??????????????????????????1B??????????128????

??

??

????????????????????????????

```
/* ??????????? */
from datasets import load_dataset
raw_data = load_dataset("json", data_files="dataset.json")
/* ????????? */
from transformers import AutoTokenizer
tokenizer = AutoTokenizer.from_pretrained(
```

```
    "google/gemma-3-1b-it"
)

/* ?????????? */

def preprocess(sample):
    sample = sample["prompt"]+ "\n" + sample["completion"]
    tokenized = tokenizer(
        sample,
        max_length=128,
        truncation=True,
        padding="max_length"
    )
    tokenized["labels"] = tokenized["input_ids"].copy()
    return tokenized

data = raw_data.map(preprocess)

/* ??????????????? */

from peft import LoraConfig, get_peft_model, TaskType
from transformers import AutoModelForCausalLM
import torch

model = AutoModelForCausalLM.from_pretrained(
    "google/gemma-3-1b-it",
    device_map="cuda",
    torch_dtype=torch.float16,
    attn_implementation='eager'
)

/* LORA??? */
lora_config = LoraConfig(
    task_type=TaskType.CAUSAL_LM,
    target_modules=["q_proj", "k_proj", "v_proj"]
)

model = get_peft_model(model, lora_config)

/* ???????? */

from transformers import TrainingArguments, Trainer

train_args = TrainingArguments(
    num_train_epochs=10,
    learning_rate=3e-4,
    logging_steps=25,
    fp16=True
)

trainer = Trainer(
    args=train_args,
    model=model,
    train_dataset=data["train"]
)
```

default watermark

```
/* ???????? */
trainer.train()

/* ??????????????????????? */

trainer.save_model("./trained")
tokenizer.save_pretrained("./trained")

?????1.0??????????3.0?????????????

??????????????????????????????????????

from transformers import pipeline

ask_llm = pipeline(
    model="./trained",
    tokenizer="./trained",
    device="cuda"
)

print(ask_llm("?????")[0]["generated_text"])

????????????????????????????????????????????????????
```

default watermark

Category

1. ?????
2. ???

Tags

1. AI
2. Linux
3. LLM
4. Lora
5. QLoRA
6. ??????????
7. ????
8. ????????
9. ???

Date Created
2025?8?3?
Author
kazuo-tsubaki